

IN THE CLAIMS

Please **ADD** claims 2-93 as follows. A status of the claims is provided below.

1. (Original) A system for maximizing collaborative productivity of knowledge workers,
having at least one component to:
logically decentralize a collaborative information process of knowledge workers;
logically centralize a collaborative information product of knowledge workers;
and
continuously reconcile the decentralized collaborative information process and the centralized collaborative information product.
2. (New) A computer implemented method of messaging within an access control model, comprising the steps:
creating a message so that a sender of the message has a set of access rights to the message; and
in the access control model, sending the message, wherein a receiver of the message obtains a second set of access rights to the message, such that that the second set of access rights are derived from the set of access rights of the sender of the message; and
pushing the message to the receiver so that the receiver exercises the second derived set of access rights in accessing the message.
3. (New) The computer implemented method of claim 2, further comprising maintaining the message in a secure information sharing network having a single and secure information context, so that users enter into the secure and trusted context to access the message.
4. (New) The computer implemented method of claim 2, wherein the message is an e-mail message and the e-mail message and associated e-mail service become an extension of the access control model.

5. (New) The computer implemented method of claim 2, wherein the message is an instant message and the instant message and instant messaging service become an extension of the access control model.

6. (New) The computer implemented method of claim 2, wherein the message is at least one of a threaded discussion entry and the threaded discussion and the threaded discussion service become an extension of the access control model.

7. (New) The computer implemented method of claim 2, wherein the sender is a software component and the receiver is a software component, so that programmatic messaging becomes an extension of the access control model.

8. (New) A computer implemented method of providing relative access control, comprising the steps:

receiving from a first principal, a request to create a set of derived access rights to an entity for a second principal, wherein the entity has an owner with a set of owner access rights and the first principal has a set of access rights, such that the set of derived access rights derive the set of access rights and the set of access rights derive the set of owner access rights;

ensuring that the first principal has a derivation right in the set of access rights to create derived access rights to the entity for the second principal;

when the first principal has the derivation right, creating a set of derived access rights for the second principal and encoding a derivation relation between the set of access rights of the first principal and the created set of derived access rights for the second principal, such that (a) the created set of derived access rights is at least a subset of the set of access rights, and (b) the created set of derived access rights are derived from the access rights of the owner of the entity.

9. (New) The computer implemented method of claim 8, wherein the step for receiving is a result of the first principal sending the entity to the second principal as a message so that the message is sent as an expression of derived access control.
10. (New) The computer implemented method of claim 9, wherein the message is at least one of an e-mail message and an attachment.
11. (New) The computer implemented method of claim 8, further comprising the step of unifying e-mail and a file system by creating an access control model providing relative access control by access derivation so that the file system includes e-mail as an extension and expression of the access control model.
12. (New) A computer implemented method of access evolution for automating the sharing of access rights in a computer environment having one or more access sharing relationships, comprising the steps of:
 - a) creating an access sharing relationship in which an access provider having a first set of access rights shares at least a subset of the set of access rights with an access receiver having a second set of access rights, so that the receiver's second set of access rights evolves as an automated function of the first set of access rights by way of the shared subset, wherein the access sharing relationship is one of one or more access sharing relationships in a network of access control;
 - b) defining the subset based on constraints of the access sharing relationship; and
 - c) sharing the subset with the access receiver,wherein the steps of defining and sharing comprise an iteration of access evolution.
13. (New) The computer implemented method of claim 12, wherein the sharing is implemented by derived access control, such that the access receiver derives access from the provider.

14. (New) The computer implemented method of claim 12, wherein the access sharing relationship is a first access sharing relationship further comprising creating a second access sharing relationship in the computer environment, wherein the access receiver of the first access sharing relationship is also the access provider in the second access sharing relationship.
15. (New) The computer implemented method of claim 12, wherein the access sharing relationship is a first access sharing relationship further comprising creating a second access sharing relationship in the computer environment, wherein the access provider of the first access sharing relationship is also the access receiver in the second access sharing relationship.
16. (New) The computer implemented method of claim 12, further comprising creating another access sharing relationship that is symmetrical to the access sharing relationship, such that an access provider in the another access sharing relationship is the access receiver in the access sharing relationship and an access receiver in the another access sharing relationship is the access provider in the access sharing relationship, creating a bi-directional access sharing relationship.
17. (New) The computer implemented method of claim 12, wherein the step for sharing shares access to at least one of an entity and a file.
18. (New) The computer implemented method of claim 12, further comprising evolving the access rights of the receiver over time in response to changes in the access rights of the provider.
19. (New) The computer implemented method of claim 12, wherein the access rights include access constraints.

20. (New) The computer implemented method of claim 12, wherein the step of defining the subset includes removing from the subset another subset of access, wherein the another subset contains access for which a manage exclusion is applied, so that the evolution is deferred by the manage exclusion for a limited or unlimited period time during serial workflow.
21. (New) The computer implemented method of claim 12, wherein the step of defining the subset includes adding to the subset another subset of access, wherein the another subset contains access for which a manage exclusion is released, so that access automatically evolves as a result of the completion of an iteration of serial workflow.
22. (New) The computer implemented method of claim 12, wherein the access sharing relationship is defined between autonomous peers in a distributed network.
23. (New) The computer implemented method of claim 22, wherein the access sharing relationship is defined between one company and another company.
24. (New) The computer implemented method of claim 12, further comprising deferring an iteration of access evolution for some limited or unlimited period of time by a manage exclusion, providing serial workflow.
25. (New) The computer implemented method of claim 12, executing parallel workflow if access evolution is not deferred for some limited or unlimited period of time by a manage exclusion, providing parallel workflow.
26. (New) The computer implemented method of claim 12, wherein the access provider has access rights to a newer version of the entity and the access receiver has access rights to an older version of the entity and deferring the sharing of the access rights to the newer version of the entity with the access receiver for a limited or unlimited period of time, so that the access receiver has access to the older version of the entity

prior to releasing the manage exclusion, and upon releasing the manage exclusion, the access receiver receiving access to the newer version.

27. (New) A computer implemented method for changing an entity over time using evolving data system relational algebra, comprising the steps of:

changing an entity by an adding of a relationship at a point in time, wherein the entity is a member of the relationship, so that the adding is a change;

changing the entity by a negating of the relationship at a second point in time, such that the second point in time is the same as or after the first point in time and such that the negating comprises an additive inverse of the adding so that the negating is another change; and

wherein changing includes the adding or the negating of the relationship, such that the change or the another change is encoded independent of and outside of the entity, providing continuity in a temporally evolving data structure of entities in a computer environment.

28. (New) The computer implemented of claim 27, wherein the changing is interpretable as information related to the entity.

29. (New) The computer implemented method of claim 27, wherein the changing becomes the basis for information management, such that the changing is provided to a subscribed consumer of the changing entity as information.

30. (New) The computer implemented method of claim 27, further comprising driving information synchronization and incremental updating as a function of the changing.

31. (New) The computer implemented method of claim 27, further comprising maintaining a value of the entity without mutation after creation.

32. (New) The computer implemented method of claim 27, further comprising maintaining a value of the entity without deletion after creation.
33. (New) The computer implemented method of claim 27, further comprising the step of maintaining the entity within a computer system having a plurality of entities, so that a past state of the computer system at a past point in time is included in the current state of the computer system, and there is continuity between the past state and the current state provided by evolution of the changing.
34. (New) The computer implemented method of claim 27, accumulating a set of changes as a continuous historical record of the entity within the computer environment.
35. (New) The computer implemented method of claim 27, further comprising maintaining the changing within a Complex Data Medium (CDM).
36. (New) The computer implemented method of claim 27, wherein the adding includes recording the time at which the relationship is started, at which time the adding is effective.
37. (New) The computer implemented method of claim 27, wherein the negating includes the time at which the relationship is suspended, at which time the negating is effective.
38. (New) The computer implemented method of claim 27, further comprising the step of managing either changing systematically by recording the adding and negating of the relationship among one or more entities of one or more classes, applications and forms so that the changing is managed independent of a specific class or application governing the entity.

39. (New) The computer implemented method of claim 27, further comprising the step of externalizing the change as its own entity by creating or modifying at least one other entity, including the relationship, so that the relation between the entity and the any other entity changes by the adding of the relationship rather than updating the entity itself.

40. (New) The computer implemented method of claim 27, further comprising the step of externalizing the another change as its own entity by creating or modifying at least one other entity, including the relationship, so that the relation between the entity and other entities changes by the negating of the relationship rather than updating the entity itself.

41. (New) The computer implemented method of claim 27, wherein the relationship is an n-tuple, the n-tuple containing a set of entities such that the entity is a member of the n-tuple and is identified by a relationship role.

42. (New) The computer implemented method of claim 27, wherein the change is addition or negation of a relationship.

43. (New) The computer implemented method of claim 27, further comprising encoding the change as another entity.

44. (New) The computer implemented method of claim 27, further comprising encoding the change as the relationship.

45. (New) The computer implemented method of claim 27, further comprising encoding the change including a time of addition associated with the relationship.

46. (New) The computer implemented method of claim 27, further comprising encoding the change including a time of negation associated with the relationship.

47. (New) The computer implemented method of claim 27, further comprising computing the relational composition of the entity at another point in time by integrating a set of relationships changing the entity, such that a member of the set of relationships is included in the relational sum if the member has been added at a time prior to the another point in time without being negated up to the another point in time.

48. (New) The computer implemented method of claim 27, further comprising changing data within the entity in addition to changing the entity through the adding and negating of the relationship.

49. (New) The computer implemented method of claim 48, wherein the step for changing data within the entity includes inserting, updating or deleting at least one of a field, an attribute, a property, a content and a metadata belonging to the entity.

50. (New) The computer implemented method of claim 27, further including classifying the entity, comprising the steps of:

creating a form that defines a relational structure for all instances of a class of entities, wherein the relational structure is defined by a set of relationship forms which are given names in the form such that a relationship form is an instance of another form; and

creating the entity for which its relational composition is defined by the form, such that the entity is a member of the entity class defined by the form;

51. (New) The computer implemented method of claim 27, further encoding either changing as at least one entity by creating the at least one entity including the relationship so that the changing is referential and the changing is a first-class object.

52. (New) The computer implemented method of claim 51, further comprising the step of creating another relationship linking the another entity, so that the another relationship links the another entity with other entities.

53. (New) The computer implemented method of claim 27, further comprising creating an activity entity, wherein the another relationship links the another entity with the activity entity, so that the activity entity provides a container of work wherein the work includes a set of change.
54. (New) The computer implemented method of claim 27, further encoding a structure of evolving content by the evolving data relational algebra, wherein the entity is one of a file or a document or a message or another entity containing content.
55. (New) The computer implemented method of claim 27, further encoding a structure of evolving access by the evolving data relational algebra, wherein the entity is one of an access control or access relationship or access group or access principal or manage exclusion or information context or security context or security policy or file retention policy or file purging policy or another entity governing access control and information security.
56. (New) The computer implemented method of claim 27, further encoding a structure of evolving workflow by the evolving data system relational algebra including serial workflow and parallel workflow.
57. (New) The computer implemented method of claim 27, further comprising linking the entity with at least one version of the entity via the relationship, such that the at least one version of the entity is itself an entity and so that the entity may have a set of versions.
58. (New) The computer implemented method of claim 57, further comprising the step of assigning access rights to one of the at least one version, providing version-level access control, so that access evolution operates within the version-level access control.

59. (New) The computer implemented method of claim 27, wherein the entity is a document object including at least one of a document, a subdocument, a document edit including at least one of an insertion and deletion, an XML document, an XML element, an XML attribute and an XML node.

60. (New) The computer implemented method of claim 59, wherein the relationship is a parent-child relationship, further comprising the step of encoding a containment relationship between a parent document object and a child document object.

61. (New) The computer implemented method of claim 60, wherein the changing entity is a parent document object, so that the step for adding places a child document object in the parent-child relationship with the parent document object, wherein the adding constitutes the insertion of a child node.

62. (New) The computer implemented method of claim 60, wherein the changing entity is a parent document object and the step for negating removes a document object from the parent-child relationship with the parent document object, wherein the negating constitutes the removal of a child node.

63. (New) The computer implemented method of claim 60, wherein the child document object is an insertion in content of the parent document object and further comprising the step of linking the child document object with a range of content that is zero or more characters in length in the document, so that the range is overwritable by content of the child document object.

64. (New) The computer implemented method of claim 60, wherein the child document object is a deletion in content of the parent document object and further comprising the step of linking the child document object with a range of content that is zero or more characters in length in the document, so that the range is overwritable by an empty string.

65. (New) A computer implemented method of information, comprising the steps:
managing a changing entity in a computer system, wherein a change is an addition or negation of a relationship that relates to the changing entity; and
subscribing a consumer of information related to the entity; and
pushing the change as information to the subscribed consumer when the change to the entity occurs.

66. (New) The computer implemented method of claim 65, further including bi-directional information-driven computing, comprising the steps of:
a) processing the change as information by the subscribed consumer; and
b) producing another change by changing the entity or another entity in response to the processing; and
c) pushing the another change to other subscribed consumers of the another change so that the other subscribed consumers can process the another change by repeating step a and step b,
wherein the subscribed consumer receives the pushed information and also pushes another information so that a bi-directional information-driven dynamic exists among the subscribed consumer and the other subscribed consumers.

67. (New) The computer implemented method of 66, further comprising the step of creating a dynamic view, wherein the dynamic view includes the receiving of pushed information and the pushing of information by the subscribed consumer.

68. (New) The computer implemented method of claim 65, further comprising the step of governing the subscribed consumer by a network operating system, wherein the subscribed consumer is a software component.

69. (New) The computer implemented method of claim 68, further comprising activating the software component to process the change.

70. (New) The computer implemented method of claim 68, further comprising deactivating the subscribed component after processing the change.

71. (New) The computer implemented method of claim 68, further comprising preventing the change by throwing an exception by the software component, resulting in the prevention of the change and dependent changes so that information consistency is maintained among a decentralized network of software components that are processing the change, wherein the decentralized network includes the component.

72. (New) A computer implemented method for delivering relational information, comprising the steps of:

tracking an entity as it is being actively consumed at a point in time by a subscribed consumer;

selecting based upon criteria contextual entities that are related by one or more relationships to the entity; and

pushing the selected contextual entities as information to the subscribed consumer, so that the subscribed consumer receives contextual information relating to the entity the subscribed consumer is processing.

73. (New) A computer implemented method of claim 72, further comprising pushing the information in a context bar.

74. (New) A computer implemented method of claim 73, further comprising providing selectable actions in the context bar that the subscribed consumer can take on the entity.

75. (New) A computer implemented method of claim 72, wherein the entity is at least one of a document, a subdocument, a document content insertion and a document content deletion contained by a document.

76. (New) A computer implemented method for linking at least two entities in a semantic web, comprising the steps of:
- a) creating at least one relationship linking the at least two entities;
 - b) automating the at least one relationship, allowing the automation to respond to interaction with at least one of the at least two entities by a subscribed consumer of the at least one of the at least two entities; and
 - c) pushing information about the at least one relationship within a computer environment to a subscribed consumer of at least one of the at least two entities.
77. (New) The computer implemented method of claim 76, further comprising the step of automating the active relationship, pushing to the subscribed consumer the information about the semantic web before allowing the subscribed consumer to update one of the at least two entities, so that the semantic web is capturing, managing, and protecting the content meaning and context of the at least two entities.
78. (New) The computer implemented method of claim 76, further comprising the step of locking the at the least two entities for at least one subscribed consumer, wherein the locking includes preventing other subscribed consumers from changing the at least two entities or providing the other subscribed consumers the information about the semantic web so that the locking and the providing enables the subscribed consumers to maintain semantic consistency and correctness of the least two entities.
79. (New) The computer implemented method of claim 78, wherein the at least two entities are at least two document objects, so that meaning and context of the at least two document objects is encoded by the semantic web.
80. (New) The computer implemented method of claim 79, wherein the one of the at least two documents objects is an insertion object, further comprising the step of linking the insertion object to a range of content in the document that is zero or more characters in length that is overwritable by content of the insertion object.

81. (New) The computer implemented method of claim 79, wherein the one of the least two documents objects is a deletion object, further comprising the step of linking the deletion object to a range of content in the document that is zero or more characters in length that is overwritable by an empty string.

82. (New) The computer implemented method of claim 76, further comprising aggregating work performed by users in the semantic web, wherein the semantic web belongs to an activity and the activity serves as an organizational unit of work.

83. (New) The computer implemented method of claim 82, wherein the work is the at least two entities and wherein that the at least two entities are document changes and wherein a document change is an insertions or deletion.

84. (New) The computer implemented method of claim 82, further comprising selecting a working activity so that one or more changes made by a user are recorded as part of the working activity.

85. (New) A computer implemented method for unified relational messaging, comprising the steps:

creating a first message as a discussion thread among one or more consumers so that the one or more consumers communicate in relation to the discussion thread;

creating a second message related to the discussion thread, wherein a relationship links the first message with the second message creating a relational context for the first message and the second message; and

pushing the second message to the one or more consumers of the first message facilitating a structured discussion context among the one or more consumers.

86. (New) The computer implemented method of claim 85, further comprising replying to the first message, wherein the second message is a reply to the first message and wherein the relationship is a containment relationship assigning the first message as a

parent in the containment relationship and assigning the reply message as a child in the containment relationship.

87. (New) The computer implemented method of claim 85, further comprising forwarding the first message, wherein the second message is a forward of the first message and wherein the relationship is a containment relationship assigning the first message as a child in the containment relationship and assigning the forward of the message as a parent in the containment relationship;

88. (New) The computer implemented method of claim 85, further comprising the step of evolving access, wherein the one or more consumers receive evolving access to at least one of the second message and the first message.

89. (New) The computer implemented method of claim 85, further including commenting on an another entity comprising the step of linking the another entity with the message by another relationship so that the first message provides a comment related to the entity.

90. (New) The computer implemented method of claim 89, wherein the another entity is a content entity contained by a document.

91. (New) A complex data medium (CDM) comprising a data structure defining one or more entities, the data structure further defining for at least one of the one or more entities a relationship among the one or more entities so that a change in the evolving composition of the one or more entities is recordable as the addition or negation of the relationship at a point in time of the change.

92. (New) A computer program product comprising a computer useable medium having readable program code embodied in the medium, the computer program product includes at least one component to:

create a message so that a sender of the message has a set of access rights to the message; and

in the access control model, send the message, wherein a receiver of the message obtains a second set of access rights to the message, such that that the second set of access rights are derived from the set of access rights of the sender of the message; and

push the message to the receiver so that the receiver exercises the second derived set of access rights in accessing the message.

93. (New) A computer program product comprising a computer useable medium having readable program code embodied in the medium, the computer program product for changing an entity over time using evolving data system relational algebra and includes at least one component to:

change an entity by an adding of a relationship at a point in time, wherein the entity is a member of the relationship, so that the adding is a change;

change the entity by a negating of the relationship at a second point in time, such that the second point in time is the same as or after the first point in time and such that the negating comprises an additive inverse of the adding so that the negating is another change; and

wherein change includes the adding or the negating of the relationship, such that the change or the another change is encoded independent of and outside of the entity, providing continuity in a temporally evolving data structure of entities in a computer environment.